Title: CMOS TAPERED GATE AND SYNTHESIS METHOD

FIELD OF THE INVENTION:

This invention relates to CMOS logic synthesis and in particular to logic synthesis in high-frequency CMOS designs.

Background:

It is common practice to specify the logic description of a CMOS design in a high-level language (such as Verilog or VHDL) and to synthesis this description into a circuit level implementation. Synthesis selects gates from a discrete gate library. It is especially common to synthesis random control logic to reduce the time to develop CMOS designs. Unfortunately synthesized circuit implementations are often slower than a non-synthesized (custom) circuit implementation and these synthesized control logic paths often limit the speed of high-frequency CMOS designs.

Summary of the Invention:

The disclosed tapered gate and synthesis methodology improves the quality of synthesized implementations. The critical path delays of these new implementations much closer to the delays of a custom circuit implementation. The discrete gate library is augmented with tapered gates to give synthesis more freedom in generation of a circuit implementation. In a tapered gate, the widths of the stacked devices are varied to achieve significant input pin to output pin delay differences. For example the bottom device(s) in a stack are designed with longer widths than the top device(s) to achieve smaller top input to output pin delay at the expense of larger bottom input to output pin delay. New synthesis algorithms are developed to exploit these tapered gates. Tapered and non-tapered gates are functionally equivalent--they differ in delay characteristics only. The input to output pin delay characteristics are coded in a rule. A timing analysis routine invokes these rules to compute arrival times and slacks (timing criticality) for each net in an implementation. A sorted, by timing criticality, list of nets is provided to the tapered gate synthesis algorithm.

The gate library from which gates are selected comprises a set of non-tapered gates and a set of tapered gates. The non-tapered gates are characterized by a stack of devices of the same width and the tapered gates are charactered by a stack of devices of different widths. Also, for each non-tapered gate there exists a plurality of tapered gates which are gunctionally equivalent to the non-tapered gate. Each set of tapered gates includes NAND gates, NOR gates, AND-OR-INVERT gates, and OR-AND-INVERT gates.

This algorithm modifies the input net to gate pin connections and swaps traditional non-tapered gates with tapered gates to improve the delay of the most timing critical paths. The latest arriving gate input net is swapped with the net connected to the top pin. The gate is then temporarily converted to a tapered gate and the timing analysis routine is invoked to re-compute arrival times and slacks for all nets. This tapered gate is retained if the slack of the temporary implementation is better than the slack of the original design.

These and other improvements are set forth in the following detailed description. For a better understanding of the invention with advantages and features, refer to the description and to the drawings.

Description of the Drawings:

Figure 1 illustrates a non-tapered 3-input CMOS NAND gate.

Figure 2 illustrates a tapered 3-input CMOS NAND gate.

Figure 3 illustrates a non-tapered 2-input CMOS NOR gate.

Figure 4 illustrates a tapered 2-input CMOS NOR gate.

Figure 5 is a graph of the input to output delay characteristics of a tapered 3-input NAND gate.

Figure 6 is a flow of the tapered gate synthesis algorithm in accordance with my preferred embodiment of the method.

My detailed description explains the preferred embodiments of my invention, together with advantages and features, by way of example with reference to the drawings.

DETAILED DESCRIPTION OF THE INVENTION:

Any CMOS gate containing a stack of devices with height greater than one can be tapered. CMOS NAND gates contain a two or higher stack of NFET devices. CMOS NOR gates contain a two or higher stack of PFET devices and CMOS AIR and OAI gates contain (two or higher) stacks of both NFET devices and PFET devices. The only common CMOS gate which cannot be tapered is an inverter since it simply consists of a (1-high) NFET device stack and a (1-high) PFET device stack. A tapered gate is illustrated with a 3-input NAND and 2-input NOR.

Figure 1 illustrates the devices comprising a non-tapered 3-input CMOS NAND gate. The PFET devices 10, 11 and 12, have same width, PW, and the NFET devices 13, 14, and 15, have same width, NW. It is common knowledge to those skilled in the art of CMOS design, that the widths of the devices comprising a gate determine the gates delay characteristics. In particular the beta ratio (PW/NW) determines the rising input and falling input delay characteristics.

Figure 2 illustrates the devices comprising a tapered 3-input CMOS NAND gate. Note this tapered gate is functionally equivalent to the 3-input NAND in Figure 1. The PFET devices 20, 21 and 22, have same width, PW, the NFET device 23 has width NW, the NFET device 24 has width t*NW and the NFET device 25 has width u*NW. The value of parameters t and u strongly influence the rising input delay characteristics of the gate. In particular consider the case where t and u are both greater than 1. The delay from rising top input pin A (26) to falling output node Y (29) is reduced since the bottom NFET devices 24 and 25 are wider compared to the non-tapered gate. The wider device widths effectively reduce the resistance of the NFET stack which speeds up the discharging of the output node 29 from Vdd to ground. But the path delay through input pin C (28) to output node Y (29) is increased. This is not obvious from a stand-alone analysis of this gate. But consider the gate which drives input pin C; the tapered gate with u>1 has larger input capacitance and this increases the delay of the gate driving input pin C. Thus, it is clear that the parameters t and u can be varied to change the input delay characteristics of the gate.

Figure 3 illustrates the devices comprising a non-tapered 2-input CMOS NOR gate. The NFET devices 30 and 31 have same width, NW, and the PFET devices 33 and 34 have same width, PW. It is common knowledge to those skilled in the art of CMOS design, that the widths of the devices comprising a gate determine the gates delay characteristics. In particular the beta ratio (PW/NW) determines the rising input and falling input delay characteristics.

Figure 4 illustrates the devices comprising a tapered 2-input CMOS NOR gate. Note this tapered gate is functionally equivalent to the 2-input NOR in Figure 3. The NFET devices 40 and 41 have same width, NW, the PFET device 43 has width PW and the PFET device 44 has width t*PW. The value of parameter t strongly influences the falling input delay characteristics of the gate.

In particular consider the case where t is greater than 1. The delay from falling input pin A (46) to rising output node Y (49) is reduced since the top PFET device 44 is wider compared to the non-tapered gate. The wider device width effectively reduces the resistance of the PFET stack and speeds up the charging of the output node Y from ground to Vdd. But the path delay through input pin B (48) to output node Y (49) is increased. This is not obvious from a stand-alone analysis of this gate. But consider the gate which drives input pin B (48); the tapered gate with t>1 has larger input capacitance and this increases the delay of the gate driving input pin B. Thus, it is clear that the parameter t can be varied to change the input delay characteristics of the NOR gate.

Other types of CMOS gates may be tapered using the same methods just described for the 3-input NAND and 2-input NOR. These types of gates include (but are not limited to) CMOS NAND gates with 2 or 4 inputs, CMOS NOR gates with more than 2 inputs, and CMOS AND-OR-INVERT (AOI) and OR-AND-INVERT (OAI) gates with any number of inputs. CMOS AOI and OAI gates contain both a 2 or more high PFET stack and a 2 or more high NFET stack; therefore both the NFET and PFET stacks may be tapered for these gates.

Figure 5 is a graph which illustrates how the path delay through 3-input NAND gate pins A and C varies as taper ratio parameters t and u are varied. Note that t=u=1 corresponds to the non-tapered gate. As the taper ratio is increased the delay through pin A (26) is reduced. However increasing taper ratio causes the delay through the pin C (28) to increase. (The delay characteristics of paths through 2-input NOR gate pins A and B as parameter t is varied in similar.) It is prohibitive to provide gates with a continuum of taper ratios t and u in a discrete gate library. From the graph, it can be observed that the majority of the delay improvement through pin A is obtained with t=2 or 3. Thus, only a few discrete values of t and u would be required in a tapered gate library. Also, parameters t and u needn't be equal;

consider the tapered gate with t=1 and u>1. In this case the delay through pins A (26) and B (27) are reduced and the delay through pin C (28) is increased. Such a tapered gate would be useful to speed up the delay of two timing critical paths. Thus multiple functionally-equivalent tapered gates may exist for each type of non-tapered gate.

Figure 6 is a flowchart of the synthesis algorithm which exploits the tapered gate library. The algorithm is invoked after an initial timing analysis. The timing analysis generates a list of timing critical gate instances. In algorithm step 60, the next gate instance, G, is selected from this list: if no critical timing instances remain the algorithm terminates. Otherwise G is examined to see if it is a candidate for tapering in step 61. If G is an inverter (which cannot be tapered) the algorithm returns to step 60 otherwise the timing criticality of the nets connected to the input of G are examined in step 62. If the most timing critical net, N, is not connected to pin A of G the algorithm swaps the net connected to pin A with net N (step 63). If net N is already connected to pin A no swapping is necessary. The algorithm then enters step 64 where G is replaced with the next functionally equivalent tapered gate G'. Timing analysis is invoked again in step 65 to re-compute the timing criticality of the paths through G', if timing is improved (step 66) then the tapered gate G' is retained and the algorithm returns to step 60. If timing is not improved G' is replaced with the original gate G (step 67) and the algorithm returns to step 64. If all functionally equivalent tapered gates have been evaluated (step 64) the algorithm returns to step 60.

While the preferred embodiment to the invention has been described, it will be understood that those skilled in the art, both now and in the future, may make various improvements and enhancements which fall within the scope of the claims which follow. These claims should be construed to maintain the proper protection for the invention first described.